**Week 3 Update**

**4 Aug 2022**

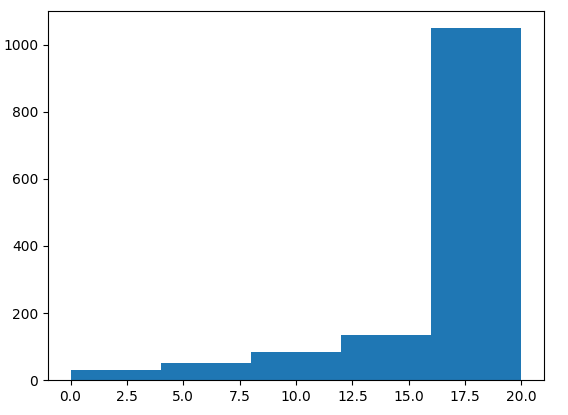
**Outcomes**

* **Extracted sequences** of student slide completions up until the third week (w3p2). These represent the Xs in the data.
* **Computed scores** for each student and their sequences by summing the number of Ps (slides passed) in the sequence e.g. a sequence of the form 1000PP 0011PP has a score of 4.
  + Note that the highest possible score that can be achieved using sequences up to the third week is 20. This is because there are 5 modules – w1p1, w1p2, w2p1, w2p2, w3p2 that have interactive slides, and there are 4 problems in each module. w3p1 seems to have some issues – to be investigated later.
* **Plotted the distribution of scores among students**
* **Trained and evaluated two regressors** 
  + **Regressors**
    - Decision tree (max. depth = 5)
    - MLP regressor (max. iterations = 500)
  + **Training and testing split**
    - **Dataset size:** 803. There are ~4000 students, but only 803 have attempted all problems up until the third week (w3p2).
    - **Training set:** 33% of dataset
    - **Testing set:** 67% of dataset

**Results**

**Distribution of scores among students**

The distribution of scores among students is left-skewed, with most students achieve high range scores (16+). If this were converted into a grade classification, this would represent most students achieving a score of 80% or Distinction and above.

****

**Regressors**

* Decision tree (max. depth = 5) – R2 score of -0.06
* MLP regressor (max. iterations = 500) – R2 score of 0.002
* Other performance measures: MAE, MSE
* We need to be realistic about how much performance to expect from such a system, esp. given the data is left-skewed.
* Discretise – predict the range

[0 0 01010101001011] [score out of 20]

Taking the sum of all 1s – plot that vs score. Linear regression model that you could think of, that would give you a sense of whether correlates with scores.

Because the data is so skewed – we know that people do tend to complete modules, it is heavily skewed towards completion. Only 16% of people failed a problem tend to skip the next one. Broadly, most people complete each thing perfectly. It’s hard for a model to outperform a basic one. Doesn’t lend itself to a more sophisticated ML model.

Recommendations:

1. Plot sum of 1s vs scores
2. Work out which slides are important
3. how consistent are people
4. Could try changing the outcome metric – number of attempts taken to submit rather than simple P/F for problems. Histogram demonstrates it’s not a particularly predictable thing – not a lot of diversity in outcomes.
   1. Could change outputs to have other things in it
      1. Number of attempts
      2. Number of failed problems
      3. Number of non-attempted problems
   2. Could try outputting tuples of those
5. Look at the number of submissions made before passing
6. Look at sequencing – they make an attempt, but it doesn’t pass, what do they go next to look at.
7. Try looking at prediction at the module level (do it module by module). Take the data for that module – input space is the sequence of activities for that particular module. Then you would need to simplify data in some way because the modules are not homogeneous.
8. What else could we predict
   1. Previously, we thought about it as a clustering task. Finding clusters of behaviour. From end user behaviour, would be useful to identify struggling students. Look at pattern of behaviours. Struggling student – failed test vs not attempted test.
   2. We could take activity for previous problem and predict whether or not they will attempt the next task – one of the things that will be most predictive is if they failed the previous problem. This could be a good way to work it out. Reduce predictive task to did they attempt the next module. If we find out the most important predictor as they failed the previous problem or low engagement overall, or low slide completion.
   3. Then we could do prediction task for if they failed the problem.
   4. Do it with data from previous module or total history?
      1. If we reduce it down to statistics for each module, then it becomes something more translatable between modules e.g. proportion of slides attempted as opposed to which slides were attempted, e.g. number of attempts at a problem, final completion status for the problems (+1 for if they pass both, 0.5 if they pass one).
      2. Could do statistics for previous module (do this first) and statistics for further ones.
      3. But check the data first

Paper

1. That would be of pedagogical interest – previous predictive task

Secondary recommendations:

1. Process mining – sequence of activities that a person does in an attempted module. They look at some slides, try completing some of the slides. There is a sequence of activities. If you could mine those common groups of processes, that might be a way of looking at diff. behaviours. This is a secondary task.

Slide completion – 0s 1s

Temporal encoding – must be careful, it would be a big shift to include timing. Restrict investigation right now to just sequencing.

We’re looking at modules up to wk3 – 5 modules up to wk3/mid-way point in the course

For each module, there are 4 problems 5 x 4 = 20 problems (max score of 20.)